In presenting the dissertation as a partial fulfillment of the requirements for an advanced degree from the Georgia Institute of Technology, I agree that the Library of the Institute shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish from, this dissertation may be granted by the professor under whose direction it was written, or, in his absence, by the Dean of the Graduate Division when such copying or publication is solely for scholarly purposes and does not involve potential financial gain. It is understood that any copying from, or publication of, this dissertation which involves potential financial gain will not be allowed without written permission.

7/25/68

MAXIMAL FLOW AT MINIMAL COST THROUGH A SPECIAL NETWORK WITH GAINS

A THESIS

Presented to

The Faculty of the Division of Graduate

Studies and Research

by

Clark Weldon Smith

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Industrial Engineering

Georgia Institute of Technology

September, 1971

MAXIMAL FLOW AT MINIMAL COST THROUGH

A SPECIAL NETWORK WITH GAINS

Approved:

Chairman

Date approved by Chairman: 8/25/71

ACKNOWLEDGMENTS

To the members of my thesis reading committee I extend my sincere thanks for their counsel and thorough consideration of my work. I am most grateful to Dr. John J. Jarvis, my faculty advisor, for his guidance and the wealth of his time invested for my benefit.

My mother and sister deserve appreciation for their typing of my first draft, an extensive labor.

Finally, I wish to express my gratitude for my wife Carol. Her patience and loving understanding afforded me an atmosphere conducive to the development of this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

CHAPTER I

INTRODUCTION

One vital step in the design of a manufacutring process is the design of a quality control system. In the assembly of a multi-component product, there may be numerous inspections of the product and its subassemblies at various stages in the process. The number and location of these inspections can significantly affect the cost of production. At each inspection a fraction of the in-process material will be found defective and will be lost from the system. It is advantageous to remove a defective piece at the earliest point, as this removal will eliminate the cost of additional processing of a defective piece. The removal will also leave more assembly line capacity open to non-defective pieces. The cost of inspection, however, may outweigh these savings. In comparing alternative quality control systems, it would be highly desirous to know the maximal number of units that could be produced and the minimal cost to produce these units.

This system can be modeled and evaluated as an acyclic network flow with positive gains problem. (For a review of network flow theory see Flows in Networks by Ford and Fulkerson (8).) The model would have nodes representing portions of the manufacturing procedure and arcs representing inspections and production routing. Each arc would have a capacity, a cost, and a gain associated with it. The capacity, $c(x, y)$, of an arc is the maximal number of units that can be processed at node x. The cost, $a(x, y)$, represents the unit cost of the process at x plus the cost of inspection. The gain, $k(x, y)$, equals the expected fraction of units that will pass the inspection. The network flow with gains problem differs from the ordinary network flow problem of Ford and Fulkerson (8) in that $f(x, y)$ units of flow leaving node x on arc (x, y) results in $k(x, y)f(x, y)$ units of flow at node y. An arc which does not represent an inspection and is used only for routing will have a gain of one. A quality control system might be modeled as in Figure 1 below.

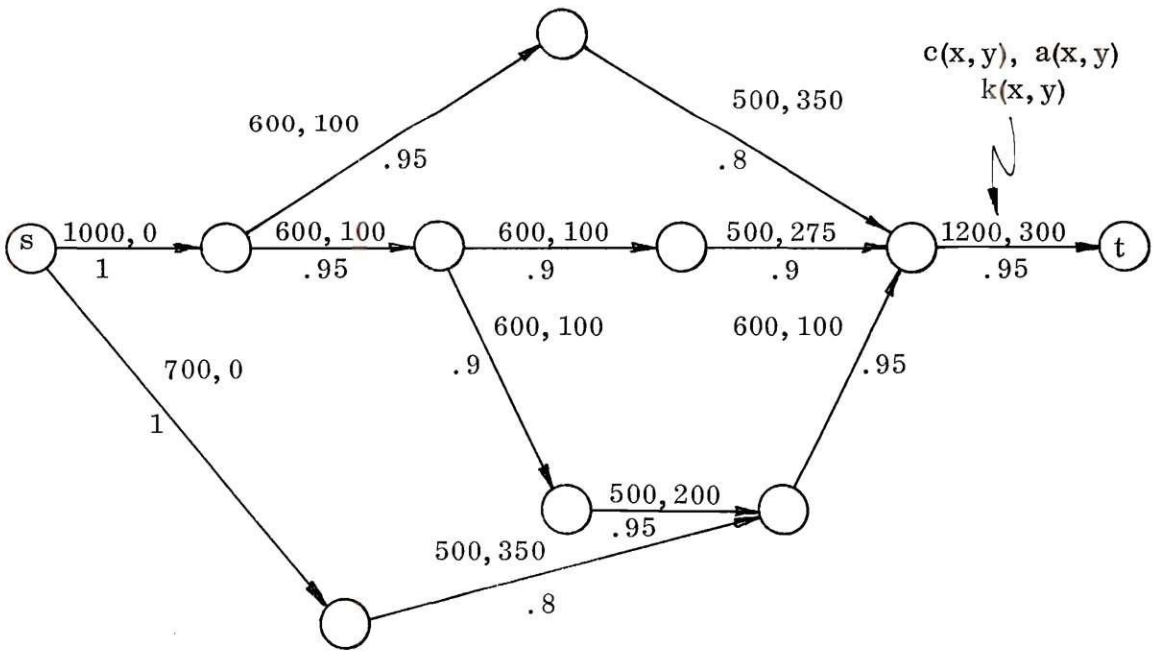

Figure 1. A Quality Control Network

The question of the maximal flow at the minimal cost in this network can be formulated as the following linear programming problem:

$$\text{(a)} \quad \text{Maximize } pv_t - \sum_{x,y} a(x, y)f(x, y) \tag{1.1}$$

Subject to: (b) $\sum_{y} [f(x,y) - k(y,x)f(y,x)] = \begin{cases} v_s & x=s \\ 0 & x \neq s,t \\ -v_t & x=t \end{cases}$

(c) $0 \leq f(x,y) \leq c(x,y) \quad \forall\ (x,y)$

(d) $v_s, v_t \geq 0$

(e) $p >> 0$

The network of Figure 1 has three additional characteristics that will be held as restraints in this paper. These additional restraints are:

1. All $k(x,y) > 0$.
2. All arcs are directed.
3. The network is acyclic.

It should be noted that in the special case of $k(x,y)=1$ for all (x,y), this problem reduces to the minimal cost flow problem discussed by Ford and Fulkerson (8). Likewise if $a(x,y)=0$ for all (x,y) in 1.1, the problem would reduce to the max-flow with gains problem. This problem with positive gains and an acyclic network was solved by Jezior (11).

Jezior (11) presents other possible applications of the acyclic network with gains. These include health care systems, multi-level maintenance systems, education systems and personnel systems.

The objective of this research is to develop and justify an efficient algorithm to determine the minimal cost, maximal flow out of the source of an acyclic network with positive gains. As an extension of this procedure an algorithm will be presented that determines the minimal cost, maximal flow into the sink of an acyclic network with positive gains. The paper will include an economic interpretation of the algorithm. The fourth chapter provides additional interpretation of Jezior's algorithms (11).

CHAPTER II

LITERATURE SEARCH

Although the literature is sparse in the area of networks with gains, valuable contributions have been made by Jewell (10), Johnson (12), and Jezior (11). There are other articles which deal with special cases of the flow with gains problem. These articles are reviewed by Jezior (11).

Jezior (11) provides algorithms for finding the maximal flow out of the source and the maximal flow into the sink in an acyclic network with positive gains. The example given by Jezior demonstrates that generally the maximal flow leaving the source does not imply the maximal into the sink. As Jezior does treat the acyclic network with positive gains, his algorithm is a convenient aid to the algorithms developed in this paper and is used almost in its entirety in both algorithms. Jezior's algorithms are included in the appendix of this paper.

Jewell (10) presents a very definitive treatment of the general flow with gains problem. Jewell's procedure is conceptually the same as the max-flow min-cost algorithm presented in this paper. Both are primal-dual procedures. At each iteration they each find the incrementally cheapest path in the network which will absorb flow from the source and maximize flow on that path. The dual variable change in both algorithms is an operation employing the dual variables of both the original problem and those of a reduced problem with no costs associated. The dual variable change in this algorithm is a direct modification of the change developed by Jewell.

Although the two algorithms are conceptually similar, there are procedural differences that make the max-flow min-cost algorithm presented here more easily implemented. These differences are enumerated below:

1. Jewell's algorithm treats the general case while the algorithm in this paper is limited to networks without cycles and with positive gains.
2. The initialization procedures are different. Jewell starts with all dual variables equal to zero. The algorithm presented here assigns the initial dual variables by the use of a least-cost procedure that identifies the least-cost path from s to t in one pass. Jewell's method will usually require one full iteration for each arc on the least-cost path before the first flow can be added to the network.
3. Jewell's algorithm requires the addition of an artificial arc to balance the flow at the sink. The algorithm presented here deals only with the original network and allows a net flow into t.
4. At each flow change a node will receive at most two labels in the algorithm of this paper. In Jewell's algorithm a node may be relabelled many times. Each time a node is relabelled, it is necessary to trace through the labelling history in order to determine whether one of the labels was derived from the other.
5. Jewell's flow change is accomplished only after two labellings back to the source for each node involved. Flow changes are carried in the labels of the

algorithm developed here.

6. Jewell only changes flow around a cycle. The max-flow min-cost algorithm of this paper can change flow around a cycle, but it can also increase flow in one pass along a path from the source to the sink without the more tedious calculations of a cycle change. Jewell accomplishes the same source to sink flow change by adjusting flow around the cycle formed by the source to sink path and the artificial arc.

Minieka (13) offers a variation of Jewell's work. Minieka suggests that Jewell's algorithm be initiated with a positive flow on the arcs. This initiation requires the addition of an artificial arc at each node. Minieka's modification proceeds to find "out-of-kilter" (8) arcs and then uses the Jewell algorithm to adjust the net output at the nodes.

Johnson (12) provides the outline of an algorithm for determining the max-flow min-cost flow in a general network with gains. This algorithm is conceptually different from the max-flow min-cost algorithm presented here. Johnson's procedure would "cost out" arcs, much as the simplex method costs out variables, to find one that could improve the current solution. This method would admit any path that would allow flow to be increased or that would lower the cost of existing flow. As has been explained earlier, the algorithm presented here adds flow to the cheapest available path.

CHAPTER III

THE MINIMAL COST FLOW WITH GAINS ALGORITHM

1. General Considerations

In this chapter an algorithm for determining the maximal flow at the minimal cost into an acyclic network with positive gains is presented and justified. As described earlier, the network (with n nodes and m arcs) will contain only directed arcs and have all $k(x, y) > 0$.

The primal problem will be of the form:

$$\text{(a)} \quad \text{Maximize } pv_s - \sum_{x,y} a(x,y)f(x,y) \tag{3.1}$$

Subject to:

$$\text{(b)} \quad \sum_{y} [f(x,y) - k(y,x)f(y,x)] = \begin{cases} v_s & x = s \\ 0 & x \neq s,t \\ -v_t & x = t \end{cases}$$

$$\text{(c)} \quad 0 \leq f(x,y) \leq c(x,y) \quad \forall \ (x,y)$$

$$\text{(d)} \quad v_s, v_t \geq 0$$

Where $p \gg 0$.

Designating P_x as the dual variables for 3.1 b. and $G_{x,y}$ for 3.1c., the dual of 3.1 is:

$$\text{(a)} \quad \text{Minimize } \sum_{x,y} c(x,y)G_{x,y} \tag{3.2}$$

Subject to: (b) $P_x - k(x,y)P_y + G_{xy} \geq - a(x,y) \qquad \forall (x,y)$

(c) $P_s \leq - p$

(d) $P_t \geq 0$

(e) $G_{x,y} \geq 0.$

Then for any set of P's the objective function 3.2a. will be minimized by setting $G_{x,y} = \max\{0, k(x,y)P_y - P_x - a(x,y)\}$. Applying this substitution and the weak theorem of complementary slackness, the optimality criteria are determined to be:

$$\text{(a)} \quad a(x,y) + P_x - k(x,y)P_y > 0 \Rightarrow f(x,y) = 0 \tag{3.3}$$

$$\text{(b)} \quad a(x,y) + P_x - k(x,y)P_y < 0 \Rightarrow f(x,y) = c(x,y)$$

$$\text{(c)} \quad v_s > 0 \Rightarrow P_s = -p$$

$$\text{(d)} \quad v_t > 0 \Rightarrow P_t = 0$$

Throughout this paper these criteria 3.3a-d will be referred to as complementary slackness conditions.

2. Discussion of the Algorithm

Jezior (11) gives a procedure to guarantee a path from s to all other x in the network. This procedure is also necessary before the algorithm is initiated and therefore is repeated here.

Step 1. Remove all arcs entering s and leaving t.

Step 2. Discard any node which has no arcs.

Step 3. Discard any node, except s, having only arcs leaving it and discard these arcs.

Step 4. Discard any node, except t, having only arcs entering it and discard these arcs.

Step 5. Repeat 2 through 4 until no change results.

The algorithm will be initiated with a primal feasible solution and an "almost feasible" dual solution, i.e. one satisfying all dual constraints except 3.2c and all complementary slackness conditions except 3.3c. This is accomplished by setting all f(x, y)=0, $P_t=0$, and assigning all other dual variables in a manner that insures that $\bar{A}(x, y) = a(x, y) + P_x - k(x, y)P_y \geq 0$ for all (x, y). This assignment of dual variables is performed in step 1 of the algorithm.

Step 2 of the algorithm assigns dual variables to the problem without costs in such a way that $\pi(x) - k(x, y)\pi(y) \geq 0$ for all (x, y). This is accomplished by using step 1 of Jezior's max-flow with gains algorithm (11). The assignment of dual variables in Step 1 and 2 is a modification of a procedure presented by Charnes and Raike (6).

The assignment of the dual variables in the first step results in the definition of the least-cost path from s to t. This path is defined by all arcs such that $\bar{A}(x, y) = 0$. The initial flow in the network is established along this least-cost path in Step 3 of the algorithm. The flow is assigned by maximizing the flow on a reduced graph consisting of all arcs with $\bar{A}(x,y) = 0$. That the path defined by the set of arcs with $\bar{A}(x,y) = 0$ is the least-cost path will be demonstrated later in this chapter.

With the flow maximized on the reduced graph, the arcs that were not included in the reduced graph are searched to find which arc or arcs can be added to provide a path for increased flow from the source at the minimal additional cost. These arcs are found by the calculation of a quantity δ in step 4 of the algorithm. δ is the marginal cost of one additional unit of flow. In most cases only one arc will produce δ, but it is possible that there will be more than one path that could absorb flow at the same cost. In any case the arc or arcs that have $\dfrac{\bar{A}(x,y)}{k(x,y)\pi(y) - \pi(x)} = \delta$ are saved in a set B for addition, one at a time, to the reduced graph. The dual variable change, insures that all arcs in the cost problem will maintain complementary slackness, i.e. $\bar{A}(x,y) > 0 \Rightarrow f(x,y) = 0$, $\bar{A}(x,y) < 0 \Rightarrow f(x,y) = c(x,y)$.

Step 5 of the algorithm adds the arcs from set B, one at a time, to a new reduced graph and increases flow on the new path or around a cycle in this new graph. The step begins by defining a new reduced graph of all arcs with $\bar{A}(x,y)=0$ but not a member of set B. These are the arcs of the previous reduced graph minus the arc that became empty or saturated and broke the previous path. Next an arc from B is selected and identified.

If neither $\pi(x)$ nor $\pi(y)$ equals zero for the arc (x,y) from B then both x and y are connected to the source by two flow augmenting paths, and the flow change will be around a cycle. In this case part c. of step 5 is implemented.

Part c. is begun by labelling nodes from s in such a way as to identify the path from s to each end node of the new arc. If there is no flow on the arc, the path from s to x is the increasing path and the path to y is the decreasing path. If the arc is saturated s-x is the decreasing path and s-y is the increasing path.

Following the path defined previously in this step, a new label is given to each node on each of the paths. The labels on the increasing flow path represent the greatest increase possible at that node. The labels on the decreasing path represent the greatest decrease possible at that node. Node s will receive two labels, the maximal decrease and the maximal increase. Since flow must be conserved at every node in the cycle except s, the minimum of the maximal increase possible at x and the maximal decrease possible at x will be the maximal change possible at x. This maximal change at x is $e'(x) = \min\left\{-\dfrac{e_x(s)}{\pi(x)},\ -\dfrac{e_y(s)}{k(x,y)\,\pi(y)}\right\}$.

To see that this is true consider the path from s to x. Let R^+ be the set of forward arcs in the path and R^- the set of reverse arcs. Jezior (11), accordingly has shown that

$$\pi(x) = \frac{\prod_{R^-} k(x,y)}{\prod_{R^+} k(x,y)}\ \pi(s)$$

or since $\pi(s) = -1$

$$\pi(x) = -\frac{\prod_{R^-} k(x,y)}{\prod_{R^+} k(x,y)} \quad .$$

Then

$$-\frac{e_x(s)}{\pi(x)} = e_x(s)\ \frac{\prod_{R^+} k(x,y)}{\prod_{R^-} k(x,y)} \quad .$$

Since $e_x(s)$ is the maximal change at s to be passed along the path from s to x,

$- \frac{e_x(s)}{\pi(x)}$ is the maximal change from s at x. A similar analysis will show that

$- \frac{e_y(s)}{k(x,y)\,\pi(y)}$ is the maximal change from s at x along a path from s to y to x.

Then $e'_x(x)$ is the maximal change in flow at x that will maintain conservation of flow at x.

Part c. is completed by changing the flow around the cycle and assigning new π's. The assignment of π's is done in such a way that $\pi(x) - k(x,y)\,\pi(y) = 0$ for all (x, y) connected to s and having $0 < f(x,y) < c(x,y)$. It will be shown later that this labeling will also result in $\pi(x) - k(x,y)\pi(y) \geq 0$ for empty arcs and $\pi(x) - k(x,y)\,\pi(y) \leq 0$ for saturated arcs.

If an arc is selected from B with either $\pi(x)$ or $\pi(y)$ nonzero and the other equal to zero, arc (x, y) connects the tree rooted at s to a tree rooted at t or to a cycle. In this case part d. of step 5 is implemented. This section of the algorithm begins by reassigning the π's of the nodes connected to s by the addition of arc (x, y). If $\pi(t)$ remains equal to zero, the flow change will again be around a cycle. The cycle must first be identified by a process similar to that used to define the paths in a cycle discussed previously. The difference being that only arcs with $\pi(x) - k(x,y)\,\pi(y) = 0$ are labelled. Since there is only one cycle (this will be proven later), only one arc will have $\pi(x) - k(x,y)\,\pi(y) \neq 0$. If for this arc $\pi(x) - k(x,y)\,\pi(y) < 0$, the path from s to x is the increasing path and path from s to y is the decreasing path. If $\pi(x) - k(x,y)\,\pi(y) > 0$, s to x is the decreasing path and s to y the increasing path. The change of flow around this cycle is identical to the change made in part c.

If $\pi(t) \neq 0$ after the reassignment of π's, a flow augmenting path exists from s to t. The flow on this path is maximized following the procedure of Jezior's max-flow algorithm (11).

If an arc is selected from B with $\pi(x) = \pi(y) = 0$, the arc is saved for possible consideration after a change of π's but is ignored until such a change takes place. The fact that the π's of both nodes are zero indicates that the arc is not connected to s in the reduced graph, and therefore flow cannot be increased out of s.

The process of choosing an arc from B and following one of the procedures above continues until there are no arcs left in B or until $\pi(x) = \pi(y) = 0$ for all arcs in B. The algorithm returns to step 4 for another dual variable change in the problem with costs. The algorithm continues to oscillate between dual variable changes and flow changes until δ in step 4 equals infinity which indicates that the current solution is optimal.

3. The Algorithm

Step 1. (Initialize Node Numbers in the Cost Problem)

a. Set $f(x,y) = 0 \;\; \forall \; (x,y)$

b. Set $P_t = 0$.

c. Let $s(x) = \{y \mid y \text{ is a successor of } x\}$; i.e. there is an arc (x,y) in the network.

d. Select some x for which all $y \in s(x)$ have assigned node numbers.

Set $P_x = \max\{k(x,y)P_y - a(x,y) \mid y \in s(x)\}$.

e. Repeat d. until all nodes have assigned node numbers.

Step 2. (Initialize Node Numbers in Problem without Costs).

Follow Step 1 of Jezior (see Appendix A).

Step 3. (Establish initial flow).

a. Let $\bar{A}(x, y) = a(x, y) + P_x - k(x, y)P_y$.

b. Consider a reduced problem of all arcs with $\bar{A}(x, y) = 0$.

c. Find the maximum flow on the reduced graph using Jezior steps 2 and 3.

d. Set $\pi'(x) = 0 \;\forall\; x \in \bar{X}$.

Step 4. (Dual Variable Change in the Cost Problem).

a. Find $\delta = \min \left\{ \dfrac{\bar{A}(x, y)}{k(x, y)\,\pi(y) - \pi(x)} \;\middle|\; \text{This quotient is positive} \right\}$.

b. If the above quotient is not positive for any (x, y) then set $\delta = \infty$ and stop, the current flow is optimal.

c. Let $P'_x = P_x + \delta\,\pi(x)$

d. Determine the set of arcs B where

$$B = \left\{ (x, y) \;\middle|\; \frac{\bar{A}(x, y)}{k(x, y)\,\pi(y) - \pi(x)} = \delta \right\}.$$

e. Calculate $\bar{A}'(x, y) = a(x, y) + P'_x - k(x, y)P'_y \qquad \forall\; (x, y)$

Step 5. (Flow Change)

a. Define a new reduced graph of all nodes and all arcs (x, y) such that $\bar{A}'(x, y) = 0$ and $(x, y) \notin B$.

b. Select an arc from B and

if $\pi(x) \neq \pi(y)$ and $\pi(x) = 0$ or $\pi(y) = 0$ go to d, else;

if $\pi(x) \neq 0$ and $\pi(y) \neq 0$ go to c, else;

if $\pi(x) = \pi(y) = 0$ select another arc from B, else;

if $B = \Phi$ or $\pi(x) = \pi(y) = 0 \;\; \forall \; (x, y) \in B$ go to Step 4.

c. Flow change around a cycle

(1) Label nodes

i) Assign s, $L(s) = -$

ii) If x is labelled, y is unlabelled, and (x, y) is in the reduced graph, label y $L(y) = x^+$

If x is labelled, y is unlabelled, and (y, x) is in the reduced graph label y $L(y) = x^-$.

(2) If $f(x, y) = 0$ for the arc from B, remove the arc from B and add the arc to the reduced graph, otherwise go to (8).

i) Label x $L'(x) = [e_x(x) = c(x, y) - f(x, y)]$ continue to s by

$L'(z) = [e_x(x)]$

where if $L(x) = z^+$ $\; e_x(z) = \min\left\{c(z, x) - f(z, x), \dfrac{e_x(x)}{k(z, x)}\right\}$

if $L(x) = z^-$ $\; e_x(z) = k(x, z) \min\{f(x, z), \; e_x(x)\}$

ii) Label y $L'(y) = [e_y(y) = k(x, y) \; (c(x, y) - f(x, y))]$

continue to s by $L'(z) = [e_y(z)]$

where if $L(y) = z^+$ $\; e_y(z) = \min\left\{ f(z, y), \dfrac{e_y(y)}{k(z, y)}\right\}$

if $L(y) = z^-$ $\; e_y(z) = k(x, y) \min\{ c(y, z) - f(y, z), \; e_y(y)\}$

(3) Set $e'_x(x) = \min \left\{ -\frac{e_x(s)}{\pi(x)}, \; -\frac{e_y(s)}{k(x,y)\,\pi(y)} \right\}$

and $e'_y(y) = -\,k(x,y)e'_x(x)$

(4) For both paths compute

$$e'(x_i) = \begin{cases} \dfrac{e'(x_{i+1})}{k(x_i, x_{i+1})} & \text{if } L(x_{i+1}) = x_i^+ \\[2ex] e'(x_{i+1})k(x_{i+1}, x_i) & \text{if } L(x_{i+1}) = x_i^- \end{cases}$$

(5) The new flow values are

$$f'(x,y) = f(x,y) + e'_x(x)$$

$$f'(x_i, x_{i+1}) = f(x_i, x_{i+1}) + e'(x_i) \qquad \text{if } L(x_{i+1}) = x_i^+$$

$$f'(x_{i+1}, x_i) = f(x_{i+1}, x_i) - e'(x_i) \qquad \text{if } L(x_{i+1}) = x_i^-$$

(6) Assign new π's

i) Set $\pi(s) = -1$

ii) If $\pi(x)$ exists, arc (x,y) is admissible, and $0 < f(x,y) < c(x,y)$

then set $\pi(y) = \frac{\pi(x)}{k(x,y)}$

If $\pi(y)$ exists, arc (x,y) is admissible, and

$0 < f(x,y) < c(x,y)$ then set $\pi(x) = k(x,y)\,\pi(y)$

iii) If $\pi(x)$ exists, $\pi(y)$ does not exist, and arc (x,y) is admissible

then set $\pi(y) = \frac{\pi(x)}{k(x,y)}$, return to ii)

If $\pi(y)$ exists, $\pi(x)$ does not exist, and arc (x, y) is admissible then set $\pi(x) = k(x, y)\pi(y)$, return to ii)

iv) If $\pi(x)$ does not exist then set $\pi(x) = 0$.

(7) Return to b. above.

(8) If $f(x, y) = c(x, y)$ for the arc from B, remove the arc from B and add the arc to the reduced graph.

i) Label x $L'(x) = [e_x(x) = f(x, y)]$

continue to s by $L'(z) = [e_x(z)]$

where if $L(x) = z^+$ $\quad e_x(z) = \min\left\{f(z, x), \dfrac{e_x(x)}{k(z, x)}\right\}$

if $L(x) = z^-$ $\quad e_x(z) = k(x, z) \min\{ c(x, z) - f(x, z),\ e_x(x) \}$.

ii) Label y $\quad L'(y) = [e_y(y) = k(x, y)f(x, y)]$

continue to s by $L'(z) = [e_y(z)]$

where if $L(y) = z^+$ $\quad e_y(z) = \min\left\{c(z, y) - f(z, y), \dfrac{e_y(y)}{k(z, y)}\right\}$

if $L(y) = z^-$ $\quad e_y(z) = k(y, z) \min\{f(y, z),\ e_y(y)\}$.

(9) Set $e'_x(x) = - \min\left\{- \dfrac{e_x(s)}{\pi(x)},\ - \dfrac{e_y(s)}{k(x, y)\,\pi(y)}\right\}$

and $e'_y(y) = - k(x, y)\, e'_x(x)$

(10) Repeat (4) - (7) above

d. Flow change from s into $\bar{X}$

(1) Add the arc from B to the reduced graph and remove the arc from B.

(2) Assign node numbers by

if $\pi(x) \neq 0$, $\pi(y) = 0$ and arc (x, y) is in

the reduced graph then set $\pi'(y) = \frac{\pi(x)}{k(x, y)}$

if $\pi(y) \neq 0$, $\pi(x) = 0$ and arc (x, y) is in the reduced

graph then set $\pi'(x) = k(x, y)\,\pi(y)$.

(3) If $\pi(t) = 0$

i) Assign s. $L(s) = -$

ii) If x is labelled, y is unlabelled, and (x, y) is in the reduced

graph with $\pi(x) - k(x, y)\,\pi(y) = 0$ label y $L(y) = x^{+}$.

If x is labelled, y is unlabelled, and (y, x) is in the

reduced graph with $\pi(y) - k(y, x)\,\pi(x) = 0$

label y $L(y) = x^{-}$.

iii) Consider the arc from ii) above with $\pi(x) - k(x, y)\,\pi(y) \neq 0$:

If $\pi(x) - k(x, y)\pi(y) < 0$ go to Step 5.c.2.i)

If $\pi(x) - k(x, y)\,\pi(y) > 0$ go to Step 5.c.8.i)

(4) Otherwise, proceed as in Jezior Step 2.

(5) Set $\pi'(x) = 0 \;\forall\; x \in \bar{X}$

(6) Return to b. above.

4. Justification

The following series of Lemmas, Corollaries and Theorems show that the algorithm achieves optimality in a finite number of steps while maintaining all primal constraints and complementary slackness conditions.

Lemma 1: At each iteration P_s strictly decreases.

Proof: At each iteration $P_s' = P_s + \delta\pi(s)$.

By construction $\pi(s) = -1$ and δ is always positive

therefore $P_s' < P_s$ Q.E.D.

Lemma 2: The algorithm at all iterations is primal feasible and is "almost" dual feasible.

Proof: 1) By construction flow is always conserved at all nodes except s and t and flow is always in the closed interval $[0, c(x, y)]$.

2) At the first step $P_t = 0$. For other iterations $P_t' = P_t + \delta\pi(t)$ but $\pi(t) = 0$ after the final dual variable change is actually made. Therefore $P_t' = P_t = 0$ at each iteration.

3) By defining $G_{x,y} = \max\{0, k(x, y)P_y - P_x - a(x, y)\}$, $G_{x,y} \geq 0$ and $P_x - k(x, y)P_y + G_{x,y} \geq -a(x, y)$ will be satisfied for any set of P's. Q.E.D.

Lemma 3: Complementary slackness conditions are maintained at each iteration.

Proof: 1) At the initialization step $f(x, y) = 0 \;\forall\; (x, y)$ and $\bar{A}(x, y) \geq 0 \;\forall\; (x, y)$, i.e. complementary slackness is present at initialization.

2) Flow is changed only on arcs with $\bar{A}(x, y) = 0$ so that no violations of complementary slackness occurs during flow changes.

3) For all nodes $P_x' = P_x + \delta\pi(x)$ with $\delta = \min\left\{\dfrac{\bar{A}(x,y)}{k(x, y)\pi(y) - \pi(x)} \,\middle|\, \delta > 0\right\}$.

We must now show that complementary slackness is maintained for all possible cases at the dual change. For any arc:

$$\bar{A}'(x,y) = a(x,y) + P_x' - k(x,y)P_y'$$

$$= a(x,y) + P_x + \delta\pi(x) - k(x,y)P_y - k(x,y)\delta\pi(y)$$

$$= a(x,y) + P_x - k(x,y)P_y + \delta[\pi(x) - k(x,y)\pi(y)]$$

$$= \bar{A}(x,y) + \delta[\pi(x) - k(x,y)\pi(y)]$$

Case 1: An arc such that $\dfrac{\bar{A}(x,y)}{k(x,y)\pi(y) - \pi(x)} = \delta$

$$\bar{A}'(x,y) = \bar{A}(x,y) + \delta[\pi(x) - k(x,y)\pi(y)] = \bar{A}(x,y) + \frac{\bar{A}(x,y)}{k(x,y)\pi(y) - \pi(x)}[\pi(x) - k(x,y)\pi(y)] = 0$$

Case 2: $\bar{A}(x,y) > 0$ and not case 1.

a) $[\pi(x) - k(x,y)\pi(y)] < 0$

then $\dfrac{\bar{A}(x,y)}{-[\pi(x) - k(x,y)\pi(y)]} > \delta > 0$

or $\bar{A}(x,y) > -\delta[\pi(x) - k(x,y)\pi(y)]$

which yields

$$\bar{A}'(x,y) = \bar{A}(x,y) + \delta[\pi(x) - k(x,y)\pi(y)] > 0.$$

b) $\pi(x) - k(x,y)\pi(y) > 0$

then $\delta[\pi(x) - k(x,y)\pi(y)] > 0$

and $\bar{A}'(x,y) = \bar{A}(x,y) + \delta[\pi(x) - k(x,y)\pi(y)] > 0$

c) $[\pi(x) - k(x,y)\pi(y)] = 0$

then $\bar{A}'(x,y) = \bar{A}(x,y) > 0$

Case 3: $\bar{A}(x,y) < 0$ and not case 1.

a) $\pi(x) - k(x,y)\pi(y) < 0$

then $\delta[\pi(x) - k(x,y)\pi(y)] < 0$

and $\bar{A}'(x,y) = \bar{A}(x,y) + \delta[\pi(x) - k(x,y)\pi(y)] < 0$

b) $\pi(x) - k(x,y)\pi(y) > 0$

then $\dfrac{\bar{A}(x,y)}{-[\pi(x)-k(x,y)\pi(y)]} > \delta$

$\bar{A}(x,y) < -\delta[\pi(x) - k(x,y)\pi(y)]$

$-\bar{A}(x,y) > \delta[\pi(x) - k(x,y)\pi(y)]$

so that $\bar{A}'(x,y) = \bar{A}(x,y) + \delta[\pi(x) - k(x,y)\pi(y)] < 0$

c) $\pi(x) - k(x,y)\pi(y) = 0$

$\bar{A}'(x,y) = \bar{A}(x,y) < 0$

Case 4: $\bar{A}(x,y) = 0$

a) $\pi(x) - k(x,y)\pi(y) < 0$

This implies that f(x, y) = c(x, y) from the max-flow algorithm.

$$\bar{A}'(x,y) = \bar{A}(x,y) + \delta[\pi(x) - k(x,y)\pi(y)] < 0 \Rightarrow f(x,y) = c(x,y).$$

b) $\pi(x) - k(x,y)\,\pi(y) > 0$

This implies that f(x, y) = 0 from the max-flow algorithm.

$$\bar{A}'(x,y) = \bar{A}(x,y) + \delta[\pi(x) - k(x,y)\,\pi(y)] > 0 \Rightarrow f(x,y) = 0.$$

c) $\pi(x) - k(x,y)\,\pi(y) = 0$

This implies that $0 \leq f(x,y) \leq c(x,y)$ from the max-flow algorithm.

$$\bar{A}'(x,y) = \bar{A}(x,y) = 0.$$

We have shown that the problem starts with complementary slackness and maintains it through flow changes and dual variable changes in the cost problem. Q.E.D.

Jezior (11) showed that at any iteration of the maximal flow algorithm with gains in an acyclic network a forest can be constructed from variables strictly between their upper and lower bounds. That this is not true in the minimal cost problem can be seen in the example in Figure 2 which contains an undirected cycle of arcs with flow strictly between its bounds in its optimal solution. Cycles of this type that occur before the final solution are dealt with in Step 5b. of the minimal cost algorithm.

Lemma 4: A connected component with one unique cycle has a number of arcs, m, exactly equal to the number of nodes, n.

Proof: From Berge (3) we see that a graph possesses a unique cycle if and

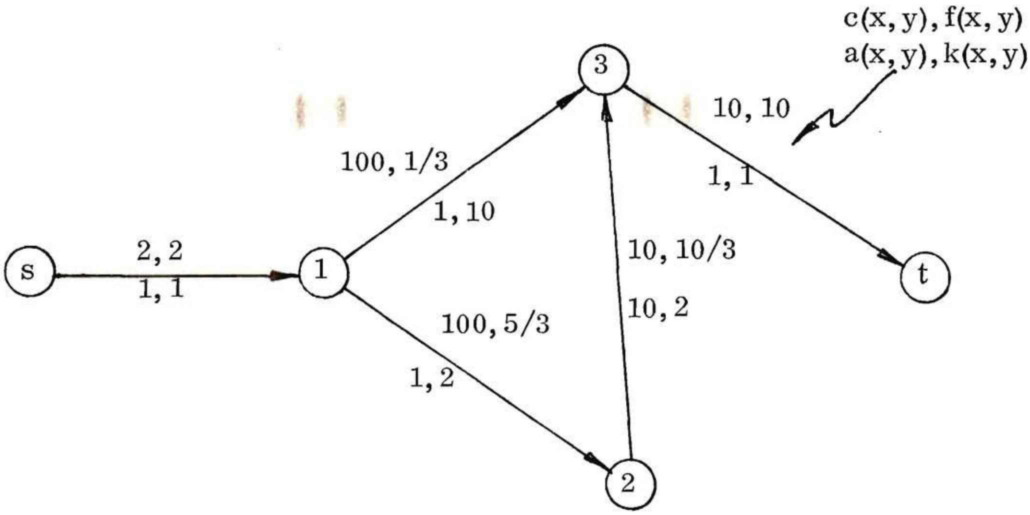

Figure 2. An Optimal Solution Containing a Cycle

only if the cyclomatic number $\nu(G) = 1$. This cyclomatic number is defined by Berge (3) as $\nu(G) = m - n + p$ where graph G has m arcs, n nodes, and p connected components. Then with one connected component ($p = 1$) and one unique cycle ($\nu(G) = 1$), $m = n$. Q.E.D.

Lemma 5: A basic solution exists if the graph of all arcs (x, y) with $0 < f(x, y) < c(x, y)$ form at least two trees and no connected component with more than one unique cycle exists.

Proof: Consider a graph of n nodes and m arcs. A basic solution will contain at most $n = m$ nonzero variables

Each the m arcs will yield at least one nonzero variable, and each arc (x, y) with $0 < f(x, y) < c(x, y)$ will place two variables in the basis at every iteration.

Then for a basic solution to exist there must be at most n-2 arcs with their flow strictly between their upper and lower bounds.

From Ford & Fulkerson (8) we see that a tree with k nodes has precisely k-1 arcs. Combining this result and the result of Lemma 4 we see that the number of arcs in a graph with at least two trees and no connected component with more than one unique cycle is less than or equal to n-2. Q.E.D.

Lemma 6: At each iteration of the max-flow min-cost algorithm, a basic solution exists.

Proof: The first iteration finds the maximum flow on a subgraph. Jezior (11) showed that such a solution can always be written in the form of a forest for an acyclic network. More precisely the arcs with flow strictly between their upper and lower bounds form a tree rooted at s and a tree rooted at t.

At the second and each succeeding iteration one of the following three cases (see Figure 3) will occur.

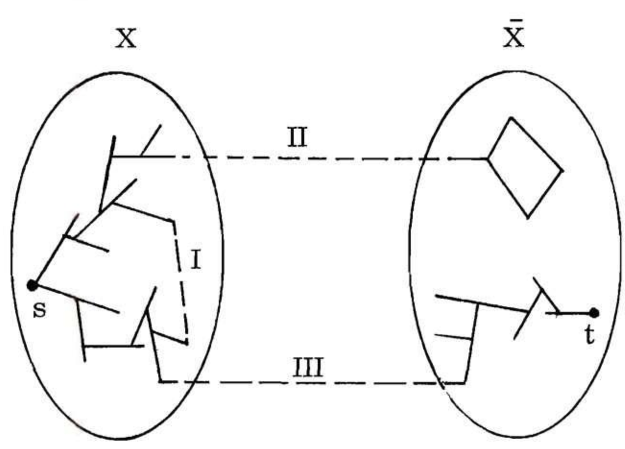

Figure 3. Possible Positions of Entering Arcs.

Case I: An arc (x, y) with $x \in X$ and $y \in X$ becomes admissible. Since the arcs in X form a tree rooted in s, the new arc (x, y) will give a total of k arcs for the k nodes. Johnson (12) showed that this implies exactly one cycle.

The algorithm would then change flow around the cycle and out of s until the cycle was broken or the cycle became disjoint from s.

Case II. An arc (x, y) with $x \in X$ and $y \in \bar{X}$ or $x \in \bar{X}$ and $y \in X$ becomes admissible and joins the tree in X to a cycle in $\bar{X}$. Again this allows one cycle.

The algorithm would then change flow around the cycle and out of s until the cycle was broken or the cycle became disjoint from s.

Case III: An arc (x, y) with $x \in X$ and $y \in \bar{X}$ or $x \in \bar{X}$ and $y \in X$ becomes admissible and joins the tree in X to the tree rooted in t in $\bar{X}$.

The algorithm would then apply Jezior's max-flow algorithm to the path s to t and the result would be a tree rooted in s and a tree rooted in t.

In these three cases the result is always two trees and single cycles that are disjoint from s and t. Applying Lemma 5 we see that this implies the existence of a basic solution.

It should be noted that these are the only three cases possible since an arc (x, y) with $x \in \bar{X}$ and $y \in \bar{X}$ can never become admissible. That such an arc can never become admissible is evident since

$$\pi(x) = 0 \;\forall\; x \in \bar{X} \text{ and } \bar{A}'(x, y) = \bar{A}(x, y) + \delta[\pi(x) - k(x, y)\,\pi(y)]. \qquad \text{Q.E.D.}$$

Theorem 1: The algorithm for maximal flow at minimal cost with gains in an acyclic network achieves an optimal solution in a finite number of steps.

Proof: Lemma 6 ensures that basic feasible solutions consisting of two trees and no connected component with more than one cycle exist at each iteration. Lemma 1 shows that P_s strictly decreases at each iteration. Since there are a finite number of basic feasible solutions; the algorithm terminates in a finite number of steps. Q.E.D.

Lemma 7: The assignment of π's after a flow change around a cycle results in:

1. $\pi(x) - k(x,y)\,\pi(y) \geq 0 \quad \forall\,(x,y) \quad$ such that $f(x,y) = 0$

2. $\pi(x) - k(x,y)\,\pi(y) = 0 \quad \forall\,(x,y) \quad$ such that $0 < f(x,y) < c(x,y)$

3. $\pi(x) - k(x,y)\,\pi(y) \leq 0 \quad \forall\,(x,y) \quad$ such that $f(x,y) = c(x,y)$

Proof: The second condition is true by construction. In the flow change, two paths are defined and flow is increased on one and decreased on the other until some arc becomes saturated or flowless. This can happen in one of four ways:

1. A forward arc on the increasing path becomes saturated.
2. A reverse arc on the increasing path becomes flowless.
3. A forward arc on the decreasing path becomes flowless.
4. A reverse arc on the decreasing path becomes saturated.

The increasing and decreasing paths are identified by examining the arc on the cycle with $\pi(x) - k(x,y)\,\pi(y) \neq 0$. If $\pi(x) - k(x,y)\,\pi(y) < 0$ for this arc the path from s to x is the increasing path. This is true as flow must be increased on arc (x, y) to satisfy the complementary slackness condition that

$$\pi(x) - k(x,y)\,\pi(y) < 0 \Rightarrow f(x,y) = c(x,y).$$

Consider the path R_1 from s to x. Let R_1^+ be the set of forward arcs in R_1 and R_1^-, the set of reverse arcs. Then as Jezior (11) demonstrated

$$\pi(x) = \frac{\prod_{R_1^-} k(x_{i+1}, x_i)}{\prod_{R_1^+} k(x_i, x_{i+1})} \pi(s) = - \frac{\prod_{R_1^-} k(x_{i+1}, x_i)}{\prod_{R_1^+} k(x_{i+1}, x_i)} .$$

Similarly if R_2 is the path from s to y and if R_2^+ and R_2^- are defined in a similar manner

$$\pi(y) = - \frac{\prod_{R_2^-} k(x_{i+1}, x_i)}{\prod_{R_2^+} k(x_i, x_{i+1})}$$

Then since $k(x, y) > 0 \;\; \forall \; (x, y)$

$$- \frac{\prod_{R_1^-} k(x_{i+1}, x_i)}{\prod_{R_1^+} k(x_i, x_{i+1})} < -k(x, y) \frac{\prod_{R_2^-} k(x_{i+1}, x_i)}{\prod_{R_2^+} k(x_i, x_{i+1})}$$

or

$$\frac{\prod_{R_1^-} k(x_{i+1}, x_i)}{\prod_{R_1^+} k(x_i, x_{i+1})} > k(x, y) \frac{\prod_{R_2^-} k(x_{i+1}, x_i)}{\prod_{R_2^+} k(x_i, x_{i+1})}$$

A similar analysis will show that if $\pi(x) - k(x, y)\,\pi(y) > 0$ the path from s to x is the decreasing path. If R_1 is the path from s to x and R_2 is the path from s to y, then

$$\frac{\prod_{R_1^-} k(x_{i+1}, x_i)}{\prod_{R_1^+} k(x_i, x_{i+1})} < k(x, y) \frac{\prod_{R_2^-} k(x_{i+1}, x_i)}{\prod_{R_2^+} k(x_i, x_{i+1})}$$

In general the quotient of the product of gains on the reverse arcs to the product of gains on forward arcs is greater for the increasing path than for the decreasing path. This fact is true, regardless of which node is assumed to separate the increasing from the decreasing path, because $k(x, y) > 0 \;\; \forall \; (x, y)$ and because a forward arc on one path can be a reverse arc on the other.

Now consider the case where the cycle is broken by a forward arc (x, y) on the increasing path becoming saturated. Define the path from s to x as R_1 and the path from s to y as R_2. Then using the notation from above

$$\frac{\prod_{R_1^-} k(x_{i+1}, x_i)}{\prod_{R_1^+} k(x_i, x_{i+1})} > k(x, y) \frac{\prod_{R_2^-} k(x_{i+1}, x_i)}{\prod_{R_2^+} k(x_i, x_{i+1})}$$

but

$$\pi(x) = - \frac{\prod_{R_1^-} k(x_{i+1}, x_i)}{\prod_{R_1^+} k(x_i, x_{i+1})} \text{ and } \pi(y) = - \frac{\prod_{R_2^-} k(x_{i+1}, x_i)}{\prod_{R_2^+} k(x_i, x_{i+1})}$$

so the inequality can be written as

$$- \pi(x) > - k(x, y) \pi(y)$$

or $\pi(x) - k(x,y)\,\pi(y) < 0$.

This inequality satisfies the complementary slackness condition since $f(x,y) = c(x,y)$. That the other three cases hold can easily be seen by following the same reasoning. Q.E.D.

5. Economic Interpretation

The following Lemma, Theorem, and their proofs demonstrate the economic significance of the dual variables P_x and the quantities--$\bar{A}(x,y)$, $\pi(x) - k(x,y)\,\pi(y)$, and δ.

Lemma 8: The modified least-cost path algorithm applied in Step 1 terminates with the dual variable P_x of any node in the graph equalling the negative of the minimal cost to send one unit of flow from the node x through the acyclic network to the sink t.

Proof: The proof is by induction. By definition $P_t = 0$ which is of course the cost from t to t. Now assume that $P_y = -C_y$ for all $y \in s(x)$ where C_y is the minimal cost of sending one unit of flow from y to t. By construction

$$
\begin{aligned}
P_x &= \max \{k(x,y)P_y - a(x,y) \mid y \in s(x)\} \\
&= \max \{-[a(x,y) + k(x,y)C_y] \mid y \in s(x)\} \\
&= \min \{a(x,y) + k(x,y)\,C_y \mid y \in s(x)\}
\end{aligned}
$$

The proposition is proved because $a(x,y) + k(x,y)C_y$ is the minimal cost of sending one unit from x along (x, y) to the sink. Then $P_x = -C_x$. Q.E.D.

Theorem 2: At each iteration the dual variable P_x of any node in the network equals the negative of the minimal cost to send one unit of flow from the node x or equivalently to reduce the flow into x by one unit.

Proof: The proof is by induction. That the theorem holds at the first iteration is a result of Lemma 8. At the first iteration $f(x, y) = 0$ for all (x, y) and the only change of flow possible is an increase from a node to the sink. Assume that after k iterations $P_x = -C_x$ for all x where C_x is the cost of increasing the flow from x by one unit. The proof will proceed by demonstrating the economic meanings of $\bar{A}(x, y)$, $[k(x, y)\ \pi(y) - \pi(x)]^{-1}$, and δ. These terms will then be related to complete the proof of the theorem.

(1) $\bar{A}(x, y)$ is the penalty incurred in placing a unit of flow on arc (x, y). For $\bar{A}(x, y)$ less than zero, the penalty would be negative and can be conceived of as the penalty for removing a unit of flow from arc (x, y). To see that this is true consider an arc (x, y). By definition $\bar{A}(x, y) = a(x, y) + P_x - k(x, y)P_y$. Substituting $- C_x$ for P_x yields

$$\bar{A}(x, y) = (a(x, y) + k(x, y)C_y) - C_x$$

which is the cost of sending one unit of flow through (x, y) and out of y minus the minimal cost of sending a unit from x.

(2) For arcs with a least one end in the set X, $[k(x, y)\ \pi(y) - \pi(x)]^{-1}$ is equal to the potential change in $f(x, y)$ resulting from an increase of one unit in flow leaving the source. An increase in flow at the source cannot affect the flow across an arc wholly in $\bar{X}$, as this arc is separated from s by a saturated cut-set. By

construction the dual variables of the nodes in the set $\bar{X}$ are equal to zero. The nodes in the set X are connected by a tree of arcs (x_i, x_j) with $\pi(x_i) - k(x_i, x_j)\pi(x_j)=0$. For each node in X there is a unique path in this tree from s to that node. Letting R^+ be the set of forward arcs in the path and R^- the set of reverse arcs, Jezior (11) demonstrated that

$$\pi(x) = \frac{\prod_{R^-} k(x_{i+1}, x_i)}{\prod_{R^+} k(x_i, x_{i+1})} \pi(s)$$

Since $\pi(s) = -1$ by construction,

$$\pi(x) = -\frac{\prod_{R^-} k(x_{i+1}, x_i)}{\prod_{R^+} k(x_i, x_{i+1})} \tag{3.4}$$

To demonstrate the validity of the interpretation of $[k(x,y)\pi(y) - \pi(x)]^{-1}$, we must examine three cases.

Case 1: $x \in X$, $y \in \bar{X}$

In this case $\pi(y) = 0$, and $[k(x,y)\pi(y) - \pi(x)]^{-1}$ becomes simply $-\frac{1}{\pi(x)}$. By substituting from equation 3.4 we obtain

$$-\frac{1}{\pi(x)} = \frac{\prod_{R^+} k(x_i, x_{i+1})}{\prod_{R^-} k(x_{i+1}, x_i)} \; .$$

Each additional unit leaving s and progressing to x is multiplied by the gains on the forward arcs and divided by the gains on the reverse arcs. It follows that

$[k(x,y)\pi(y) - \pi(x)]^{-1}$ does indeed equal the increase in flow on (x,y) for each additional unit of flow leaving s.

Case 2: $x \in \bar{X}$, $y \in X$

In this case $\pi(x) = 0$, and $[k(x,y,)\pi(y) - \pi(x)]^{-1} = [(k(x,y)\pi(y)]^{-1}$.

Applying equation 3.4 yields

$$\frac{1}{k(x,y)\pi(y)} = -\frac{1}{k(x,y)}\left[\frac{\prod_{R^+} k(x_i, x_{i+1})}{\prod_{R^-} k(x_{i+1}, x_i)}\right]$$

The quantity in brackets is, by the reasoning used in case 1, the increase in flow at y for each unit increase at s. In order to conserve flow at y, a change of $-\frac{1}{k(x,y)}$ times this increase into y must be made in $f(x,y)$.

Case 3: $x, y \in X$

This is the most difficult case, as the flow change must take place around a cycle. The flow change will be an increase on one path and a decrease on another path. Identify these paths as

$$R_x = (s = x_0, x_1, \ldots, x_k = x)$$

$$R_y = (s = y_0, y_1, \ldots, y_k = y).$$

To determine the effect of a unit flow change originating at s, consider a change of ϵ(ϵ may be positive or negative) in $f(x,y)$. In order to conserve flow, a change of ϵ at node x from path R_x and $-k(x,y)\epsilon$ at node y from path R_y would be required. A flow change of

$$\frac{\prod_{R_x^-} k(x_{i+1}, x_i)}{\prod_{R_x^+} k(x_i, x_{i+1})}\ \epsilon \tag{3.5}$$

would be necessary at s to effect a change of ϵ at x. Likewise to obtain a change of $-k(x, y)\,\epsilon$ at y, the change at s would need to be

$$-k(x, y)\ \frac{\prod_{R_y^-} k(y_{i+1}, y_i)}{\prod_{R_x^+} k(y_i, y_{i+1})}\ \epsilon \tag{3.6}$$

Summing 3.5 and 3.6, the net change at s would be

$$\left[\frac{\prod_{R_x^-} k(x_{i+1}, x_i)}{\prod_{R_x^+} k(x_i, x_{i+1})} \quad -k(x, y) \quad \frac{\prod_{R_y^-} k(y_{i+1}, y_i)}{\prod_{R_y^+} k(y_i, y_{i+1})}\right] \epsilon\,.$$

Applying 3.4 and setting the net change at s equal to one yields:

$$[-\pi(x) + k(x, y)\ \pi(y)]\ \epsilon = 1$$

Solving for ϵ we obtain:

$$\epsilon = [k(x, y)\pi(y) - \pi(x)]^{-1}$$

which is the desired result.

(3) δ is the minimal penalty of establishing a potential flow absorbing path from s. From (1) and (2) above, $[k(x,y)\pi(y)-\pi(x)]^{-1}\bar{A}(x,y)$ is the penalty incurred in placing a unit of flow from s onto a flow absorbing path containing (x, y) and with $\bar{A}(x,y) = 0$ for all other arcs on the path.

As $\delta = \min\{[k(x,y)\pi(y) - \pi(x)]^{-1}\bar{A}(x,y) \mid \delta > 0\}$, the interpretation of δ is established. δ is restrained to be greater than zero because $[k(x,y)\pi(y) - \pi(x)]^{-1}$ $\bar{A}(x,y) < 0$ implies that no flow change is possible. $[k(x,y)\pi(y) - \pi(x)]^{-1} < 0$ indicates that a flow can be decreased on arc (x, y). $\bar{A}(x,y) > 0$, however, implies $f(x,y) = 0$ and therefore there is no possible decrease. In a similar manner, $\bar{A}(x,y) < 0$ and $[k(x,y)\pi(y) - \pi(x)]^{-1} > 0$ will not admit a flow change.

We are now ready to show that P_x is the negative of the minimal cost to send one unit of flow from x at the k+1 iteration. First consider the case of $x \in \bar{X}$. This node is still attached to the previous flow absorbing path (a cycle or a path to t) and the cost of this path has not changed. The dual variable P_x also remains unchanged as $\pi(x) = 0$ and the change is defined as $P'_x = P_x + \delta\pi(x)$.

For the special case of P_s, $\pi(s) = -1$ by construction; and δ is the penalty of establishing one additional unit from s. The new value of P'_s is $P_s + \delta\pi(x)$ or substituting $-C_s$ for P_s yields

$$P'_s = -(C_s + \delta) = -C'_s .$$

For the case of $x \in X$, the flow change can be visualized as removing one unit of flow from the path into the node, conserving flow on the unique path by which $\pi(x)$ was assigned, and increasing flow from s on the new unique flow absorbing structure. This unit removed from x would be multiplied by the gain of reverse arcs and divided by the gain of forward arcs in the path $R = (s = x_0, x_1, \ldots, x_k = x)$. To conserve flow at s a flow of

$$\frac{\prod_{R^-} k(x_{i+1}, x_i)}{\prod_{R^+} k(x_i, x_{i+1})}$$

would have to exit s on the flow absorbing path. This increase would incur a penalty of

$$\frac{\prod_{R^-} k(x_{i+1}, x_i)}{\prod_{R^+} k(x_i, x_{i+1})} \delta .$$

Applying 3.4 this penalty becomes $-\pi(x)\,\delta$. Adding this penalty to the previous cost yields $C_x' = C_x - \pi(x)\,\delta = -(P_x + \pi(x)\,\delta) = -P_x'$. This is the desired result. Q.E.D.

Corollary 1: After each dual variable change, P_s equals the negative of the cost per unit of the flow added to the network in the succeeding flow change.

This corollary is a direct result of Theorem 2 and is presented here only for emphasis. This important feature of the algorithm provides immediate and constant economic analysis of the system being modeled.

6. Maximizing Flow into t at the Minimal Cost

The problem of maximal flow at minimal cost into t is formulated as:

$$\text{(a) Maximize } pv_t - \sum_{x,y} a(x,y)f(x,y) \tag{3.7}$$

Subject to: $$\text{(b) } \sum_y [f(x,y) - k(y,x)\, f(y,x)] = \begin{cases} v_s & x = s \\ 0 & x \neq s, t \\ -v_s & x = t \end{cases}$$

$$\text{(c) } 0 \leq f(x,y) \leq c(x,y) \;\; \forall \;(x,y)$$

$$\text{(d) } v_s, v_t \geq 0$$

with $p >> 0$.

The dual of the problem is:

$$\text{(a) Minimize } \sum_{x,y} c(x,y)G_{x,y} \tag{3.8}$$

Subject to: $$\text{(b) } P_x - k(x,y)P_y + G_{x,y} \geq -a(x,y) \qquad \forall (x,y)$$

$$\text{(c) } P_t \geq p$$

$$\text{(d) } P_s \leq 0$$

$$\text{(e) } G_{x,y} \geq 0$$

Setting $G_{x,y} = \max\{0, k(x,y)P_y - P_x - a(x,y)\}$ the complementary slackness conditions are:

$$\text{(a) } a(x,y) + P_x - k(x,y)P_y > 0 \Rightarrow f(x,y) = 0 \tag{3.9}$$

$$\text{(b) } a(x,y) + P_x - k(x,y)P_y < 0 \Rightarrow f(x,y) = c(x,y)$$

(c) $v_s > 0 \Rightarrow P_s = 0$

(d) $v_t > 0 \Rightarrow P_t = p$

The algorithm proceeds as follows:

Step 1 (Initialize Node Numbers in the Cost Problem)

(a) Set $f(x, y) = 0 \ \forall (x, y)$.

(b) Set $P_s = 0$.

(c) Let $p(x) = \{y \mid y \text{ is a predecessor of } x\}$.

(d) Select some x for which all $y \in p(x)$ have assigned node numbers.

$$\text{Set } P_x = \min\left\{\frac{a(x, y)+P_x}{k(x, y)} \;\middle|\; y \in p(x)\right\} .$$

(e) Repeat d. until all nodes have assigned node numbers.

Step 2 (Initialize Node Numbers in Problem without Costs).

Follow Step 1 of Jezior's maximal flow into t algorithm (See Appendix B).

Step 3 (Establish Initial Flow)

This step is accomplished as for maximizing v_s at minimal cost.

Step 4 (Dual Variable Change in the Cost Problem).

This step is the same as Step 4 of the previous algorithm.

Step 5 (Flow Change)

This step remains the same except for the sections listed below:

c. (1) i) Assign t, $L(t) = -$

(6) i) Set π (t) = 1

d. (3) If π (s) = 0,

i) Assign t, L(t) = -

(4) Otherwise, proceed as in Step 2 of Jezior's maximal flow into t algorithm.

Justification of this extension will not be shown. The proofs concerning its justification are similar to those for the maximal flow at minimal cost from the source algorithm.

7. Example

Consider the network shown in Figure 4.

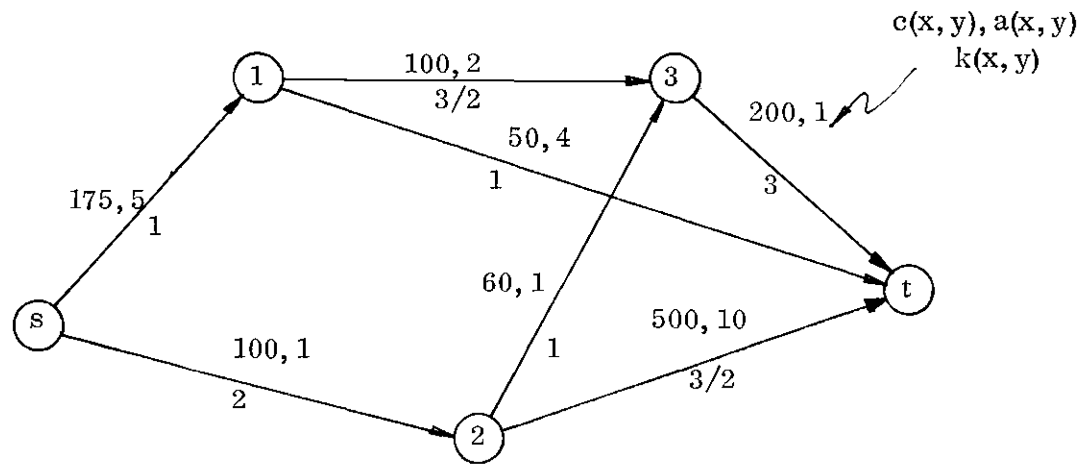

Figure 4. Example Network.

Step 1 (Initialize Node Numbers in the Cost Problem)

Setting $P_t = 0$ and solving the modified least cost path problem values for P_x are: $P_t = 0$, $P_3 = -1$, $P_2 = -2$, $P_1 = -3\frac{1}{2}$, $P_s = -5$.

Step 2 (Initialize Node Numbers in Problem without Costs)

Step 3 (Establish initial Flow)

Figure 5 shows the reduced graph with its flow and dual variables at the end of the first iteration.

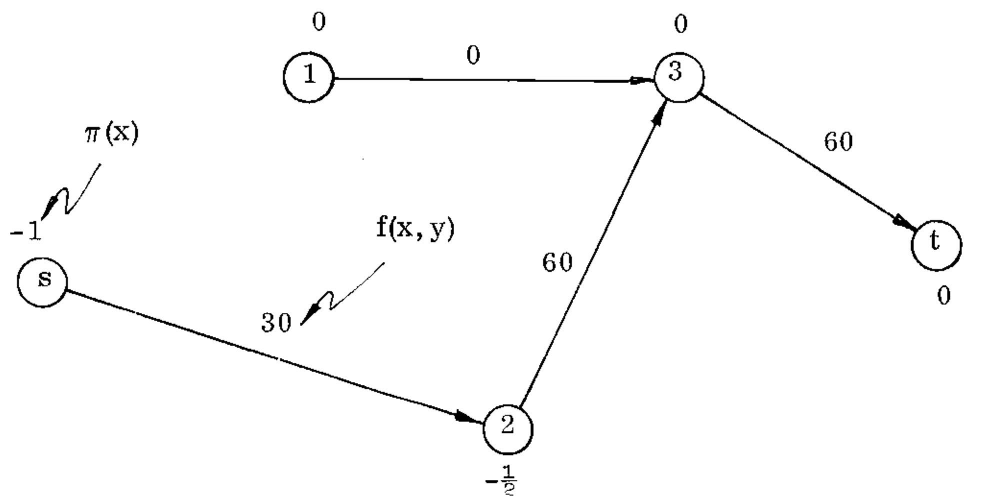

Figure 5. After the First Iteration.

Step 4 (Dual Variable Change in the Cost Problem)

Find $\delta = \min\{3\frac{1}{2}, \infty, 16\} = 3\frac{1}{2}$. Calculate the new dual variables $P_s = -8\frac{1}{3}$, $P_1 = -3\frac{1}{2}$, $P_2 = -3\ 3/4$, $P_3 = -1$, $P_t = 0$. This results in the arcs--(s, 1), (s, 2), (1, 3), and (3, 5)--being admissible.

Step 5 (Flow Change)

Flow is increased on the path s - 1 - 3 - t. The algorithm continues for five iterations to an optimal solution. Table 1 and Figure 6 show the solution and the history of the flow changes.

It should be noted that the flow change in the fourth iteration was a change around a cycle. The admissible arcs at this iteration are shown with their labels in Figure 7. At the start of the step $\pi(2) = -\frac{1}{2}$ and $\pi(3) = -2/3$. The step resulted

Table 1. History of Flow

Arc \ Iteration	0	1	2	3	4	5
f(s, 1)	0	0	93-1/3	143-1/3	150	150
f(s, 2)	0	30	30	30	25	100
f(1, 3)	0	0	93-1/3	93-1/3	100	100
f(1, t)	0	0	0	50	50	50
f(2, 3)	0	60	60	60	50	50
f(2, t)	0	0	0	0	0	150
f(3, t)	0	60	200	200	200	200

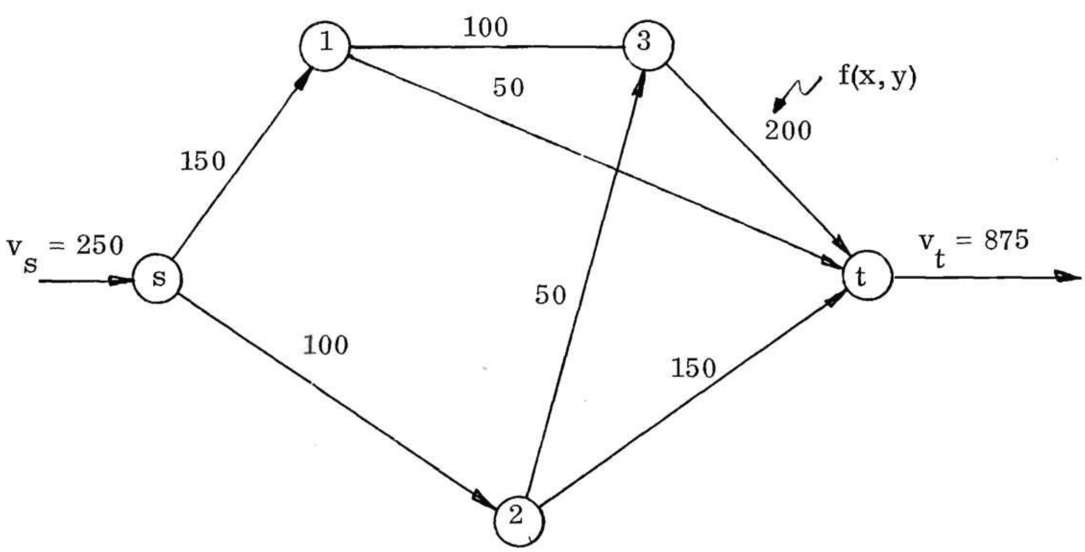

Figure 6. Optimal Flow.

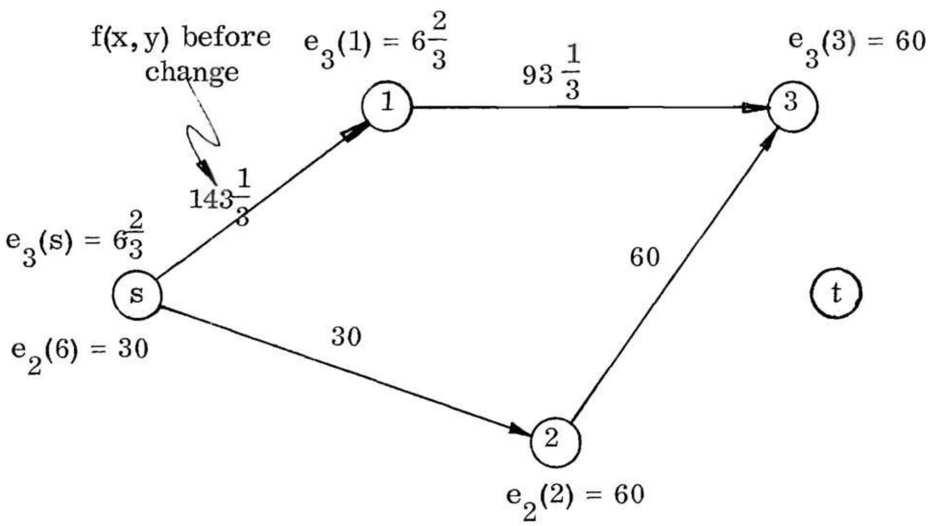

Figure 7. Flow Change Around a Cycle.

with $\Delta f(s, 1) = 6\frac{2}{3}$, $\Delta f(1, 3) = 6\frac{2}{3}$, $\Delta f(2, 3) = -10$, $\Delta f(s, 2) = -5$, $\pi(s) = -1$, $\pi(1) = -1$, $\pi(2) = -\frac{1}{2}$, $\pi(3) = -\frac{1}{2}$, $\pi(t) = 0$.

CHAPTER IV

OTHER PROPERTIES OF THE MAX-FLOW PROBLEM

In some networks it would be advantageous to obtain the maximal system output with the minimal input. For example, if production routings resulted in varying percentages of defective pieces, the most efficient system would be that system that produced the most finished goods per unit input. One could also conceive of the problem of obtaining the maximal input with the minimal output. The following two theorems demonstrate that Jezior's (11) max-flow algorithms for the acyclic network with positive gains do in fact solve the problems of max-in min-out and max-out min-in.

The problem of minimizing the flow from the source to yield the maximal flow into the sink can be formulated as:

$$\text{(a)} \quad \text{Maximize } pv_t - v_s \tag{4.1}$$

Subject to:

$$\text{(b)} \quad \sum_y [f(x,y) - k(y,x)f(y,x)] = \begin{cases} v_s & x = s \\ 0 & x \neq s,t \\ -v_t & x = t \end{cases}$$

$$\text{(c)} \quad 0 \leq f(x,y) \leq c(x,y) \quad \forall\ (x,y)$$

$$\text{(d)} \quad v_t, v_s \geq 0$$

with $p >> 0$.

The dual of 4.1 is:

$$\text{(a) Minimize } \sum_{x,y} c(x,y)\gamma(x,y) \tag{4.2}$$

Subject to: (b) $\pi(x) - k(x,y)\pi(y)+\gamma(x,y) \geq 0 \quad \forall (x,y)$

(c) $\pi(s) \leq 1$

(d) $\pi(t) \geq p$

(e) $\gamma(x,y) \geq 0$.

By taking $\gamma(x,y) = \max\{0, k(x,y)\pi(y) - \pi(x)\}$ the complementary slackness conditions are:

$$\text{(a) } \pi(x) - k(x,y)\,\pi(y) > 0 \Rightarrow f(x,y) = 0 \tag{4.3}$$

(b) $\pi(x) - k(x,y)\pi(y) < 0 \Rightarrow f(x,y) = c(x,y)$

(c) $v_t > 0 \Rightarrow \pi(t) = p$

(d) $v_s > 0 \Rightarrow \pi(s) = 1$.

Theorem 4: Jezior's algorithm (11) for determining the maximal flow into the sink of an acyclic network with positive gains achieves this flow with the minimal input from the source.

Proof: Jezior's maximal flow into t algorithm terminates with a set of dual variables that satisfy the conditions:

(a) $\pi(x) - k(x,y)\ \pi(y) > 0 \Rightarrow f(x,y) = 0$

(b) $\pi(x) - k(x,y)\pi(y) < 0 \Rightarrow f(x,y) = c(x,y)$

(c) $v_t > 0 \Rightarrow \pi(t) = 1.$

It should be noted that the algorithm terminates before the final dual variable change is made, and therefore $\pi(s) \neq 0$. In order to prove that the flow obtained by Jezior's algorithm is an optimal solution to the min-in max-out problem, we will make two dual variable changes and demonstrate that the complementary slackness conditions 4.3 are satisfied.

Step 1: Set $\pi'(x) = \dfrac{\pi(x)}{\pi(s)} \quad \forall x$

This step results in 4.3 a, b, and d being satisfied. 4.3 a and b hold because $\pi(s) > 0$ at every iteration of Jezior's algorithm.

Step 2: Set $\pi''(x) = \pi'(x) \quad \forall\ x \in X$

$$\pi''(x) = p\,\pi(s)\,\pi'(x) \quad \forall\, x \in \bar{X}$$

Where $p \gg 0$.

This step results in $\pi(t) = p$ and $\pi(s) = 1$. To see that 4.3 a and b hold consider the following cases:

(1) Arc (x,y) is wholly in X or wholly in $\bar{X}$. The conditions 4.3 a and b hold because $p > 0$.

(2) $x \in X$, $y \in \bar{X}$, $f(x,y) = c(x,y)$ and $\pi'(x) - k(x,y)\ \pi'(y) < 0$. There is no change in $\pi''(x)$ but $\pi''(y) = p\,\pi(s)\,\pi''(y) > \pi'(y)$. Therefore

$\pi''(x) - k(x,y)\pi''(y) < \pi'(x) - k(x,y)\pi'(y) < 0.$

(3) $x \in \bar{X}$, $y \in X$, $f(x,y) = 0$ and $\pi'(x) - k(x,y)\pi'(y) > 0.$

There is no change in $\pi''(y)$ but $\pi''(x) = p\pi(s)\pi'(x) > \pi'(x).$

Then $\pi''(x) - k(x,y)\pi''(y) > \pi'(x) - k(x,y)\pi'(y) > 0.$

The definition of Jezior's cut set insures that there are no other cases to be considered. Now the complementary slackness conditions 4.3 hold and the flow from Jezior's maximal into t algorithm is optimal to the min-in max-out problem 4.1. Q.E.D.

Theorem 5: Jezior's (11) algorithm for determining the maximal flow from the source of an acyclic network with positive gains achieves this flow with the minimal flow into the sink.

The proof of the theorem is highly similar to the proof of theorem 4 and will not be shown.

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

The principal result of this thesis is an efficient algorithm for determining the maximal flow at the minimal cost into an acyclic network with positive gains. The algorithm has been shown to be finite, and an economic interpretation has been presented and verified. Other results of the study are an algorithm for determining the maximal flow at the minimal cost into the sink of an acyclic network with positive gains and two theorems which show that Jezior's algorithms (11) are actually minimax procedures.

Further research is recommended in the area of flow with gains in acyclic networks. It is conjectured that the maximal flow into t is an alternate optimal solution to the maximal flow from s problem. That is, one solution exists with both v_s and v_t maximal.

As previously noted, the maximal flow into an acyclic network with gains does not imply the maximal flow out of the network. For example consider the network in Figure 8. The capacity of each arc in this network is ten (10) and the gains are given in the figure. The flow shown in the figure is the flow as assigned by Jezior's algorithm for maximal flow into the network. This solution yields 10 units into the sink. However, if flow is rerouted from (3, t) to (3, 4) and (4, t), the flow into t is 20 units.

It should also be noted that in the general network with gains problem a

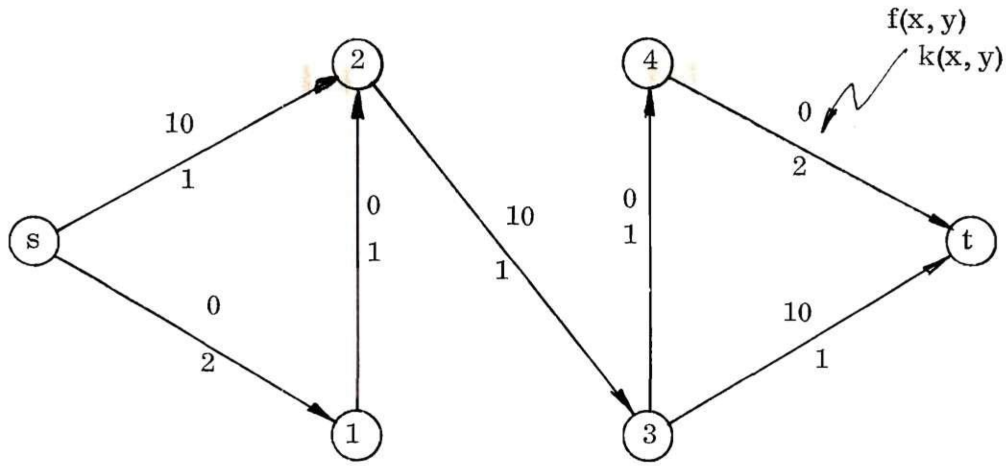

Figure 8. Max-In Does Not Imply Max-Out.

solution which has both v_s and v_t maximal need not exist. For the graph of Figure 9 the maximal input is 100 with a flow into t of zero; yet the maximal output is 50 with a flow out of s of 50.

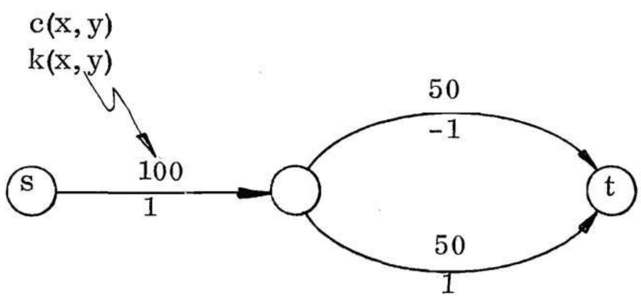

Figure 9. In General v_s and v_t can not be Maximal Simultaneously.

One possible approach to this problem is to find a solution that is optimal to both the maximal flow out of the source problem and the maximal flow into the sink problem. An algorithm is proposed that would first maximize the flow into the sink using Jezior's algorithm. The algorithm would then adjust the flow leaving s to its maximal without changing the flow into t. The proposed algorithm would proceed as follows:

Step 1. Find the maximal flow into the sink using Jezior's algorithm (see appendix).

Step 2. Assign new dual variables as follows:

(a) Set $\pi'(x) = 0 \quad \forall\ x \in \bar{X}$.

(b) Set $\pi'(s) = -1$.

(c) If $\pi(x)$ is assigned, $(x, y) \in X$, and $0 < f(x, y) < c(x, y)$

then set $\pi(y) = \dfrac{\pi(x)}{k(x, y)}$.

If $\pi(y)$ is assigned, $(x, y) \in X$ and $0 < f(x, y) < c(x, y)$

then set $\pi(x) = k(x, y)\pi(y)$.

Step 3. Increase flow out of s by:

(a) Select an arc with

$\pi(x) - k(x, y)\pi(y) > 0$ and $f(x, y) = c(x, y)$ or

$\pi(x) - k(x, y)\pi(y) < 0$ and $f(x, y) = 0$.

If there are no arcs with these characteristics, stop. The flow is optimal.

(b) Change flow following the procedure of Step 5c. of the max-flow min-cost algorithm with the one change that (7) should read:

(7) Return to Step 3a. of the max-in max-out algorithm.

While this algorithm has worked with example problems, the difficulty arises in proving that the algorithm is finite. The possibility of cycling without a flow change arises when arcs with $f(x, y) = 0$ or $f(x, y) = c(x, y)$ are in the tree of admissible arcs ($\pi(x) - k(x, y)\, \pi(y) = 0$).

APPENDIX A

JEZIOR'S MAXIMAL FLOW OUT OF s ALGORITHM

Step 1 (Initialize Node Numbers)

a. Set $\pi(s) = -1$. All other $\pi(x)$ are not yet assigned.

b. Let $p(x) = \{y \mid y \text{ is an immediate predecessor of } x\}$.

c. Select some x for which all $y \in p(x)$ have assigned node numbers.

$$\text{Set } \pi(x) = \min\left\{\frac{\pi(y)}{k(y,x)} \mid y \in p(x)\right\}$$

d. Repeat step c until all nodes have assigned node numbers.

Step 2 (Flow Change)

a. Determine admissible arcs by the criteria:

$$\pi(x) - k(x,y)\,\pi(y) = 0 \text{ where } 0 \leq f(x,y) \leq c(x,y).$$

b. Label on the network of admissible arcs as follows:

(1) Assign s, $L(s) = [-,\ e(s) = \infty]$

(2) For any node y which is unlabelled, if node x is labelled and arc (x, y) is admissible with $f(x,y) < c(x,y)$, label y with $L(y) = [x^{+},\ e(y)]$ where $e(y) = k(x,y)\min\{c(x,y) - f(x,y),\ e(x)\}$.

If node x is labelled and arc (y, x) is admissible with $f(y,x) > 0$, label node y with $L(y) = [x^{-},\ e(y)]$ where $e(y) = \min\left\{f(y,x),\ \frac{e(x)}{k(y,x)}\right\}$.

c. When t is labelled, breakthrough is achieved and a flow augmenting path is determined. The flow is assigned as follows:

Denote the flow augmenting path, P, as $s = x_0, x_1 \ldots x_k = t$. Let $e'(t) = e(t)$. For $i = k - 1 \ldots 1$, $e'(x_i)$ is;

$$e'(x_i) = \begin{cases} \dfrac{e'(x_{i+1})}{k(x_i, x_{i+1})} & \text{if } L(X_{i+1}) = \left[x_i^+, e(x_{i+1})\right] \\ e'(x_{i+1})k(x_{i+1}, x_i) & \text{if } L(x_{i+1}) = \left[x_i^-, e(x_{i+1})\right] \end{cases}$$

Flow values are:

$$f'(x_i, x_{i+1}) = f(x_i, x_{i+1}) + e'(x_i, x_{i+1}) \text{ if } (x_i, x_{i+1}) \in P$$

$$f'(x_{i+1}, x_i) = f(x_{i+1}, x_i) + e'(x_{i+1}) \text{ if } (x_{i+1}, x_i) \in P$$

All other flows remain the same.

d. Erase labels; repeat step 2b. 1. with the new feasible flow, $f'(x,y)$, until non-breakthrough occurs.

Step 3 (Dual Variable Change)

When non-breakthrough occurs:

a. Let $X = \{x \mid x \text{ is labelled}\}$ and $\bar{X} = \{x \mid x \text{ is unlabelled}\}$.

Determine sets of arcs A_1 and A_2 according to the criteria:

(a) $A_1 = \{(x,y) \mid x \in X,\ y \in \bar{X},\ \pi(x) - k(x,y)\pi(y) > 0\}$

(b) $A_2 = \{(x,y) \mid x \in \bar{X},\ y \in X,\ \pi(x) - k(x,y)\,\pi(y) < 0\}$

b. Determine θ where:

(a) $\theta = \max\{\theta_1, \theta_2, -1\}$

(b) $\theta_1 = \max\left\{\frac{\pi(x)-k(x,y)\pi(y)}{k(x,y)\pi(y)} \middle| (x,y) \in A_1\right\}$

(c) $\theta_2 = \max\left\{-\frac{[\pi(x)-k(x,y)\pi(y)]}{\pi(x)} \middle| (x,y) \in A_2\right\}$.

($\theta_i = -\infty$ if $A_i = \Phi$).

If $\theta = -1$, the solution is optimal and the algorithm is terminated. Otherwise change $\pi(x)$ as follows:

(a) $\pi'(x) = \pi(x)$, $x \in X$

(b) $\pi'(x) = (1 + \theta)\,\pi(x)$, $x \in \bar{X}$.

With $\pi'(x)$ determined, repeat Step 2.

APPENDIX B

JEZIOR'S MAXIMAL FLOW INTO t ALGORITHM

Step 1 (Initialize Node Numbers)

a. Set $\pi(t) = 1$. All other $\pi(x)$ are not yet assigned.

b. Let $s(x) = \{y \mid y \text{ is a successor of } x\}$.

c. Select some x for which all $y \in s(x)$ have been assigned node numbers.

Set $\pi(x) = \max\{k(x, y)\pi(y) \mid y \in s(x)\}$.

d. Repeat c until all nodes have been assigned node numbers.

Step 2 (Flow Change)

This step is accomplished as for maximizing v_s.

Step 3 (Node Number Change)

a. At $(X, \bar{X})$ determine A_1 and A_2 according to the criteria used in maximizing v_s.

b. Determine θ where:

$$\theta = \max\{\theta_1, \theta_2, -1\}$$

(a) $\theta_1 = \max\left\{\frac{k(x, y)\pi(y) - \pi(x)}{\pi(x)} \mid (x, y) \in A_1\right\}$.

(b) $\theta_2 = \max\left\{-\frac{[k(x, y)\pi(y) - \pi(x)]}{\pi(x)} \mid (x, y) \in A_2\right\}$.

($\theta_i = -\infty$ if $A_i = \Phi$).

If $\theta = -1$, the solution is optimal and the algorithm is terminated. Otherwise change $\pi(x)$ as follows:

(a) $\pi'(x) = (1 + \theta)\,\pi(x), \ x \in X$

(b) $\pi'(x) = \pi(x), \ x \in \bar{X}$

With $\pi'(x)$ determined, repeat Step 2.

BIBLIOGRAPHY

1. Balas, E. "The Dual Method for the Generalized Distribution Problem," Management Science, Vol 12, No. 7, 1966, pp. 555-568.

2. Balas, E. and P. Ivanescue. "On the Generalized Transportation Problem," Management Science, Vol 11, No. 1, 1964, pp. 188-203.

3. Berge, Claude. The Theory of Graphs and its Applications, John Wiley and Sons, New York, 1966.

4. Berge, C. and A. Ghouila-Houri. Programming, Games and Transportation Networks, John Wiley and Sons, New York, 1965.

5. Busacker, R. G. and T. L. Saaty. Finite Graphs and Networks, McGraw Hill, New York, 1964.

6. Charnes, A. and W. Raike. "One Pass Algorithm for Some Generalized Network Problems," Operations Research, Vol 14, No. 5, 1966, pp. 914-924.

7. Dantzig, G. B. Linear Programming and Extensions, Princeton University Press, Princeton, New Jersey, 1963.

8. Ford, L. R., Jr. and D. R. Fulkerson. Flows in Networks, Princeton University Press, Princeton, New Jersey, 1962.

9. Hu, T. C. Integer Programming and Network Flows, Addison-Wesley, Reading, Massachusetts, 1969.

10. Jezior, A. M. "An Algorithm for Maximal Flow with Gains in a Special Network," School of Industrial and Systems Engineering, Georgia Institute of Technology, December 1970.

12. Johnson, E. L. Programming in Networks and Graphs, Operations Research Center, University of California, Berkely, 1965.

13. Minieka, E. "Selected Graph Theoretic Optimization Problems," Administrative Sciences Department, Yale University, August 1970.

14. Ore, O. Graphs and their Uses, Random House, New York, 1963.